NAME

lesskey – customize key bindings for less

SYNOPSIS (deprecated)

lesskey [−o output] [−−] [input]
lesskey [−−output=output] [−−] [input]
lesskey −V
lesskey −−version

SCOPE

This document describes the format of the **lesskey** source file, which is used by **less** version 582 and later. In previous versions of **less**, a separate program called **lesskey** was used to compile the **lesskey** source file into a format understood by **less**. This compilation step is no longer required and the **lesskey** program is therefore deprecated, although the file format remains supported by **less** itself.

DESCRIPTION

A **lesskey** file specifies a set of key bindings and environment variables to be used by subsequent invocations of **less**.

FILE FORMAT

The input file consists of one or more *sections*. Each section starts with a line that identifies the type of section. Possible sections are:

#command
Customizes command key bindings.

#line-edit
Customizes line-editing key bindings.

#env Defines environment variables.

Blank lines and lines which start with a hash mark (#) are ignored, except as noted below.

COMMAND SECTION

The command section begins with the line

#command

If the command section is the first section in the file, this line may be omitted. The command section consists of lines of the form:

string <whitespace> *action* [extra-string] <newline>

Whitespace is any sequence of one or more spaces and/or tabs. The *string* is the command key(s) which invoke the action. The *string* may be a single command key, or a sequence of up to 15 keys. The *action* is the name of the less action, from the list below. The characters in the *string* may appear literally, or be prefixed by a caret to indicate a control key. A backslash followed by one to three octal digits may be used to specify a character by its octal value. A backslash followed by certain characters specifies input characters as follows:

\b	BACKSPACE	(0x08)
\e	ESCAPE	(0x1B)
\n	NEWLINE	(0x0A)
\r	RETURN	(0x0D)
\t	TAB	(0x09)

\k followed by a single character represents the char(s) produced when one of these keys is pressed:

\kb	BACKSPACE (the BACKSPACE key)
\kB	ctrl-BACKSPACE

\kd	DOWN ARROW
\kD	PAGE DOWN
\ke	END
\kh	HOME
\ki	INSERT
\kl	LEFT ARROW
\kL	ctrl-LEFT ARROW
\kr	RIGHT ARROW
\kR	ctrl-RIGHT ARROW
\kt	BACKTAB
\ku	UP ARROW
\kU	PAGE UP
\kx	DELETE
\kX	ctrl-DELETE
\k1	F1

A backslash followed by any other character indicates that character is to be taken literally. Characters which must be preceded by backslash include caret, space, tab, hash mark and the backslash itself.

An action may be followed by an "extra" string. When such a command is entered while running **less**, the action is performed, and then the extra string is parsed, just as if it were typed in to **less**. This feature can be used in certain cases to extend the functionality of a command. For example, see the "{" and ":t" commands in the example below. The extra string has a special meaning for the "quit" action: when **less** quits, the first character of the extra string is used as its exit status.

EXAMPLE

The following input file describes the set of default command keys used by **less**. Documentation on each command can be found in the man page, under the key sequence which invokes the command.

```
#command
\r          forw-line
\n          forw-line
e           forw-line
j           forw-line
\kd         forw-line
^E          forw-line
^N          forw-line
k           back-line
y           back-line
^Y          back-line
^K          back-line
^P          back-line
J           forw-line-force
K           back-line-force
Y           back-line-force
d           forw-scroll
^D          forw-scroll
u           back-scroll
^U          back-scroll
\40         forw-screen
f           forw-screen
^F          forw-screen
```

```
^V          forw-screen
\kD         forw-screen
b           back-screen
^B          back-screen
\ev         back-screen
\kU         back-screen
z           forw-window
w           back-window
\e\40       forw-screen-force
F           forw-forever
\eF         forw-until-hilite
R           repaint-flush
r           repaint
^R          repaint
^L          repaint
\eu         undo-hilite
\eU         clear-search
g           goto-line
\kh         goto-line
<           goto-line
\e<         goto-line
p           percent
%           percent
\e[         left-scroll
\e]         right-scroll
\e(         left-scroll
\e)         right-scroll
\kl         left-scroll
\kr         right-scroll
\e{         no-scroll
\e}         end-scroll
{           forw-bracket {}
}           back-bracket {}
(           forw-bracket ()
)           back-bracket ()
[           forw-bracket []
]           back-bracket []
\e^F        forw-bracket
\e^B        back-bracket
G           goto-end
\e>         goto-end
>           goto-end
\ke         goto-end
\eG         goto-end-buffered
=           status
^G          status
:f          status
/           forw-search
?           back-search
\e/         forw-search *
\e?         back-search *
n           repeat-search
\en         repeat-search-all
```

```
N           reverse-search
\eN         reverse-search-all
&           filter
m           set-mark
M           set-mark-bottom
\em         clear-mark
'           goto-mark
^X^X        goto-mark
E           examine
:e          examine
^X^V        examine
:n          next-file
:p          prev-file
t           next-tag
T           prev-tag
:x          index-file
:d          remove-file
-           toggle-option
:t          toggle-option t
s           toggle-option o
            ## Use a long option name by starting the
            ## extra string with ONE dash; eg:
            ##   s toggle-option -log-file\n
_           display-option
|           pipe
v           visual
!           shell
#           pshell
+           firstcmd
H           help
h           help
V           version
0           digit
1           digit
2           digit
3           digit
4           digit
5           digit
6           digit
7           digit
8           digit
9           digit
q           quit
Q           quit
:q          quit
:Q          quit
ZZ          quit
```

PRECEDENCE

Commands specified by **lesskey** take precedence over the default commands. A default command key may be disabled by including it in the input file with the action "invalid". Alternatively, a key may be defined to do nothing by using the action "noaction". "noaction" is similar to "invalid", but **less** will give an error beep for an "invalid" command, but not for a "noaction" command. In addition, ALL default commands may be disabled by adding this control line to the input file:

```
#stop
```

This will cause all default commands to be ignored. The #stop line should be the last line in that section of the file.

Be aware that #stop can be dangerous. Since all default commands are disabled, you must provide sufficient commands before the #stop line to enable all necessary actions. For example, failure to provide a "quit" command can lead to frustration.

LINE EDITING SECTION

The line-editing section begins with the line:

```
#line-edit
```

This section specifies new key bindings for the line editing commands, in a manner similar to the way key bindings for ordinary commands are specified in the #command section. The line-editing section consists of a list of keys and actions, one per line as in the example below.

EXAMPLE

The following input file describes the set of default line-editing keys used by **less**:

```
#line-edit
\t        forw-complete
\17       back-complete
\e\t      back-complete
^L        expand
^V        literal
^A        literal
\el       right
\kr       right
\eh       left
\kl       left
\eb       word-left
\e\kl     word-left
\ew       word-right
\e\kr     word-right
\ei       insert
\ex       delete
\kx       delete
\eX       word-delete
\ekx      word-delete
\e\b      word-backspace
\e0       home
\kh       home
\e$       end
\ke       end
\ek       up
\ku       up
\ej       down
^G        abort
```

LESS ENVIRONMENT VARIABLES

The environment variable section begins with the line

```
#env
```

Following this line is a list of environment variable assignments. Each line consists of an environment variable name, an equals sign (=) and the value to be assigned to the environment variable. White space before and after the equals sign is ignored. Variables assigned in this way are visible only to **less**. If a variable is specified in the system environment and also in a lesskey file, the value in the lesskey file takes precedence.

If the variable name is followed by += rather than =, the string is appended to the variable's existing value. This currently works only if any += lines immediately follow the same variable's original definition (with an = line), without any intervening definitions of other variables. It can append only to a variable defined earlier in the file; it cannot append to a variable in the system environment. The string is appended literally, without any extra whitespace added, so if whitespace is desired, it should be appended to the end of the preceding line. (It cannot be added to the beginning of the += string because space after the equals sign is ignored, as noted above.)

CONDITIONAL CONFIGURATION

If a line begins with #version followed by a relational operator and a version number, the remainder of the line is parsed if and only if the running version of **less** (or **lesskey**) matches the operator. This can be helpful if a lesskey file is used by different versions of **less**.

For example, suppose that a new command named 'sideways-search' is added in **less** version 777. Then the following line would assign the command to the Q key, but only in versions of **less** which support it. The line would be ignored by versions earlier than 777.

```
#version >= 777  Q sideways-search
```

These six operators are supported:

>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
=	Equal to
!=	Not equal to

The #version feature is not supported in **less** and **lesskey** before version 594. In those older versions, all #version lines are ignored.

EXAMPLE

The following input file sets the −i and −S options when is run and, on version 595 and higher, adds a −−color option.

```
#env
## (Note that there must be a space at the end of the next line,
##  to separate the --color option from the -S option.)
LESS = −i −S
#version >= 595  LESS += −−color=Hkc
```

SEE ALSO

less(1)

WARNINGS

On MS-DOS and OS/2 systems, certain keys send a sequence of characters which start with a NUL character (0). This NUL character should be represented as \340 in a lesskey file.

COPYRIGHT

AUTHOR

Mark Nudelman
Report bugs at https://github.com/gwsw/less/issues.